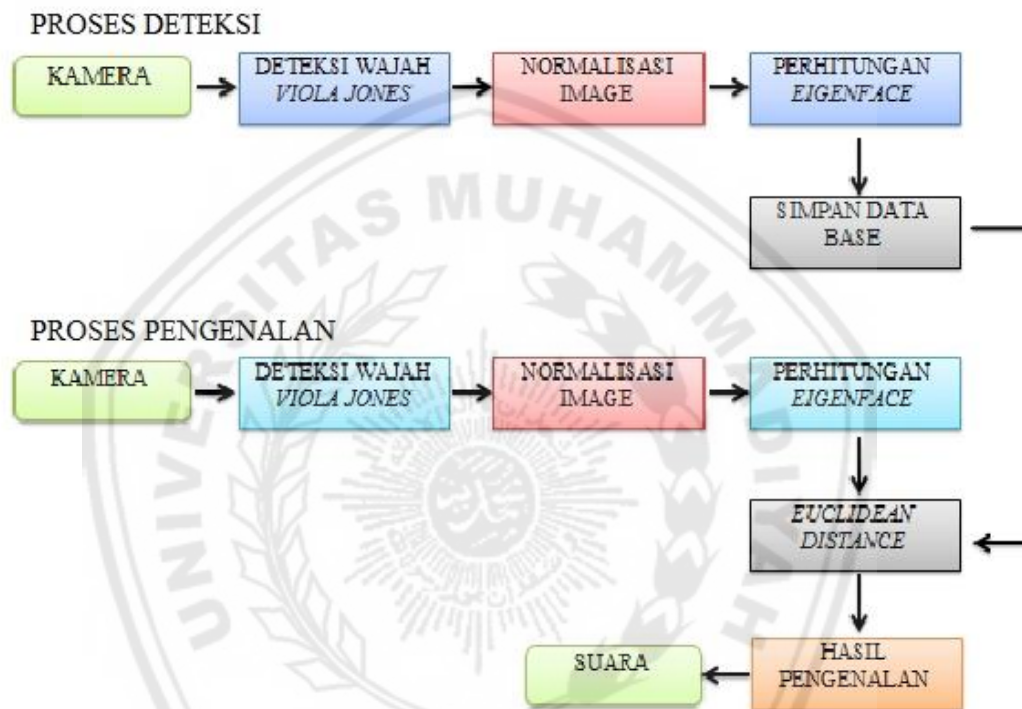


BAB III

PERANCANGAN SISTEM

Pada bab ini akan membahas tentang perancaran *hardware*, instalasi *software* sistem deteksi wajah. *Software* ini dirancang dengan harapan mampu mendeteksi dan mengenali wajah yang terdapat di dalam database dengan cepat dan akurat. Untuk blok diagram sebagai berikut:



Gambar 3.1 Diagram Blok Sistem deteksi wajah

Pada blok diagram gambar 3.1 dijelaskan. Sistem ini terdiri dari 2 blok utama yaitu, blok proses pengumpulan data karakteristik citra, proses pengenalan dan keluaran berupa suara. Blok proses pengumpulan data karakteristik terdiri dari kamera dan perangkat lunak yang memproses citra masukan menjadi karakteristik citra yang hasilnya akan disimpan. Sedangkan pada blok proses pengenalan dan *output* suara terdiri dari kamera, perangkat lunak yang memproses citra masukan menjadi karakteristik citra yang telah disimpan hingga dan memberikan *output* hasil pengenalan yang kemudian akan menghasilkan suara.

Sebuah *image* wajah baru atau *test face* akan dicoba untuk dikenali, pertama menerapkan cara pada tahapan pertama perhitungan *eigenface* untuk mendapatkan nilai *eigenface* dari *image* tersebut.

1. Aktifkan webcam untuk menampilkan gambar yang ditangkap webcam kedalam aplikasi. Pendeteksian wajah menggunakan metode Viola Jones, dilakukan secara langsung (*real time*) menggunakan webcam. Dengan *Adaptive Boosting* dan kombinasi *Cascade of Classifier* akan mempercepat pendeteksian objek wajah. Jika wajah terdeteksi, akan dilakukan penggambaran garis persegi pada wajah tersebut.
2. Penangkapan citra wajah (*image capturing*) dapat dilakukan secara langsung (*real time*) menggunakan webcam, setelah terdeteksi adanya gambar wajah dari webcam.
3. Kemudian dilakukan proses pemrosesan awal yang meliputi, normalisasi ukuran citra, RGB ke grayscale, histogram equalization untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya, resize untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input.
4. Simpan data wajah yang diambil dalam bentuk*.JPG.
5. Kemudian dilakukan proses perhitungan untuk mengutip bagian terpenting dengan metode *eigenface* sehingga didapatkan *eigenvector* dan *eigenvalue* dari gambar tersebut.
6. Proses penyimpanan kedalam data wajah untuk setiap citra wajah yang telah digunakan dalam proses perhitungan dalam bentuk *.xml , semakin kompleks dan sering maka proses pengenalan wajah akan semakin baik.
7. Data yang telah disimpan nantinya digunakan sebagai nilai pembanding pada proses penghitungan jarak untuk pengenalan wajah.
8. Kemudian dilakukan proses perhitungan untuk mengutip bagian terpenting dengan metode *eigenface* sehingga didapatkan *eigenvector* dan *eigenvalue* dari gambar tersebut.

9. Proses pengenalan wajah dengan menghitung jarak antara fitur wajah yang ada dalam data dan fitur wajah yang baru. Jarak yang didapat di cari yang terkecil untuk identifikasi.

3.1 Proses Deteksi Wajah

3.1.1 Kamera

Mengaktifkan webcam untuk menampilkan gambar yang ditangkap webcam kedalam aplikasi.

ret, img = cam.read()



Gambar 3.2 Modul webcam Raspberry

Modul kamera *raspberry* dapat terintegrasi secara langsung dengan *sistem operasi raspberry* dengan mengatur pada *port* mana *webcam* akan digunakan dan seting. Sebelum digunakan driver untuk webcam harus di instalasi karena nantinya aplikasi presensi akan melakukan koneksi ke driver webcam untuk memfungsikannya sebagai media input, jika driver sudah di instalasi maka tinggal menghubungkan webcam ke port USB dan *webcam* siap digunakan sebagai media *input* sistem Kualitas pengambilan gambar yang dapat di seting semaksimal mungkin dengan menetapkan pada spesifikasi maksimal pada *webcam*.

Pada *system*, sebagian besar di kontrol oleh *microcontroller* berupa *raspberry pi*. *Raspberry* bekerja sebagai *system* utama yang memproses file gambar dari *webcam* dan mengirim *ouput* berupa suara.

3.1.2 Deteksi Wajah Viola Jones

Pendeteksian wajah menggunakan metode Viola Jones, dilakukan secara langsung (*real time*) menggunakan *webcam*. Dengan *Adaptive Boosting* dan kombinasi *Cascade of Classifier* akan mempercepat pendeteksian objek wajah. Jika wajah terdeteksi, akan dilakukan penggambaran garis persegi pada wajah tersebut. Penangkapan citra wajah (*image capturing*) dapat dilakukan secara langsung (*real time*) menggunakan *webcam*, setelah terdeteksi adanya gambar wajah dari *webcam*.

```
faces = detector.detectMultiScale(gray, 1.3, 5)
```

```
for (x,y,w,h) in faces:
```

```
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
```



Gambar 3.3 Deteksi Wajah Viola Jones

3.1.3 Normalisasi Image

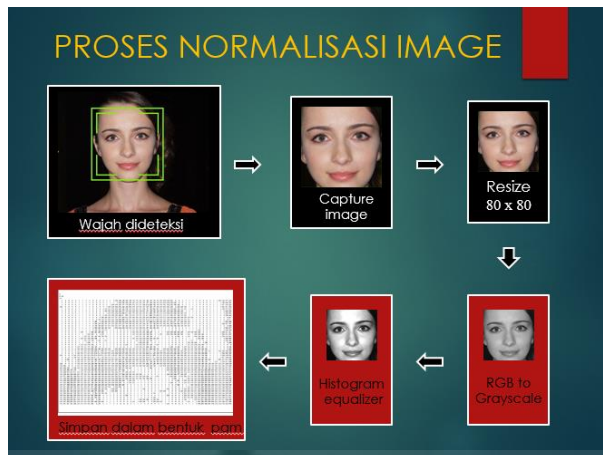
Kemudian dilakukan proses pemrosesan awal yang meliputi, normalisasi ukuran citra, RGB ke grayscale, histogram equalization untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya.

Resize untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input. Simpan data wajah yang diambil dalam bentuk *.jpg.

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
cv2.imwrite ('dataSet/User.'+Id +'.'+ str(sampleNum) + '.jpg', gray  
[y:y+h,x:x+w])
```

Seperti gambar berikut:



Gambar 3.4 Proses Normalisai Image

3.1.4 Proses EigenFace

Kemudian dilakukan proses metode eigenface untuk mengutip bagian terpenting sehingga didapatkan eigenvector dan eigenvalue dari gambar tersebut. Proses penyimpanan kedalam data wajah untuk setiap citra wajah yang telah digunakan dalam proses EigenFace dalam bentuk *.yml , semakin kompleks dan sering maka proses pengenalan wajah akan semakin baik.

```
faces,Ids = getImagesAndLabels('dataSet')
```

```
recognizer.train(faces, np.array(Ids))
```

```
recognizer.save('trainer/trainer.yml')
```

Seperti gambar berikut :

```

YAML:1.0
radius: 1
neighbors: 8
grid_x: 8
grid_y: 8
histograms:
- !!opencv-matrix
  rows: 1
  cols: 16384
  dt: f
  data: [ 2.55102036e-003, 2.55102036e-003, 0., 1.27551018e-003,
    7.65306130e-003, 0., 0., 2.16836724e-002, 0., 0., 0., 0.,
    1.27551018e-003, 0., 0., 1.78571418e-002, 6.37755077e-003, 0.,
    0., 0., 1.27551018e-003, 0., 0., 1.27551018e-003, 0.,
    2.55102036e-003, 0., 0., 3.06122452e-002, 3.82653065e-003,
    1.53061226e-002, 3.06122452e-002, 0., 0., 0., 0., 0., 0.,
    0., 0., 0., 0., 0., 0., 0., 0., 1.27551018e-003, 0.,
    0., 0., 0., 1.27551018e-003, 6.37755077e-003, 0., 0., 0.,
    8.54591802e-002, 2.55102036e-003, 3.44387740e-002,
    1.14795920e-002, 7.65306130e-003, 1.27551018e-003, 0.,
    1.27551018e-003, 2.55102036e-003, 0., 1.27551018e-003,
    1.27551018e-003, 0., 0., 0., 0., 0., 0., 0.,
    5.10204071e-003, 0., 0., 0., 1.27551018e-003, 0., 0., 0.,
    1.27551018e-003, 0., 0., 0., 1.27551018e-003, 0., 0.,
  ]
  
```

Gambar 3.5 Data Training .yml

3.2 Proses Pengenalan Wajah

Pada proses pengenalan wajah juga terdapat proses deteksi wajah viola jones, normalisasi image dan perhitungan eigenface. Sama seperti yang sudah di jelaskan pada proses deteksi wajah. Sehingga akan

3.2.1 Proses Includian Distance

Proses ini adalah proses dimana data wajah hasil proses training akan di badingkan dengan data wajah baru untuk mencari nilai terdekat. Dari wajah baru akan dicari eigen vector kemudian akan di cari nilai yang paling mendekati nilai yang tersimpan di dalam database sehingga akan didapatkan hasil Pengenalan wajah.

```
recognizer = createEigenFaceRecognizer()
recognizer.load('trainner/trainner.yml')
cascadePath = "viola_jones.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
for(x,y,w,h) in faces:
    cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
    Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
    if(conf<100):
        Id = names[ids.index(str(Id))]
```

3.2.2 Proses Suara

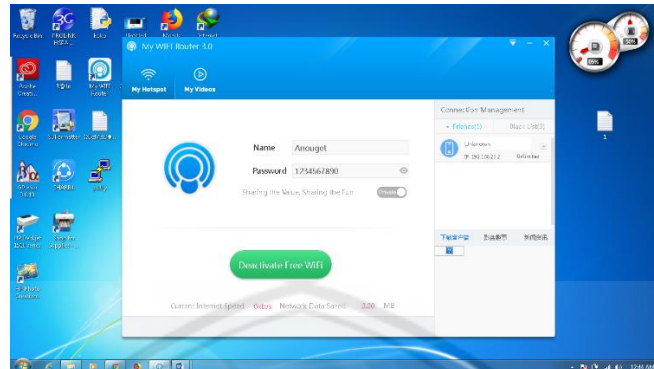
Pada proses ini wajah orang yang terdeteksis akan menampilkan id. Selanjutny id akan dirubah ke dalam bentuk suara (voice). Jika wajah seseorang tidak terdeteksi maka akan mengeluarkan suara “id tidak ditemukan” .

```
if Id!=curInd:
    eng.say(Id)
    eng.runAndWait()
    curInd=Id
else:
    Id="Unknown"
    cv2.cv.PutText(cv2.cv.fromarray(im),str(Id), (x,y+h),font, 255)
    cv2.imshow('im',im)
```


3.3 Proses Menjalankan Sistem

3.3.1 Proses Connec Raspberry pada PC

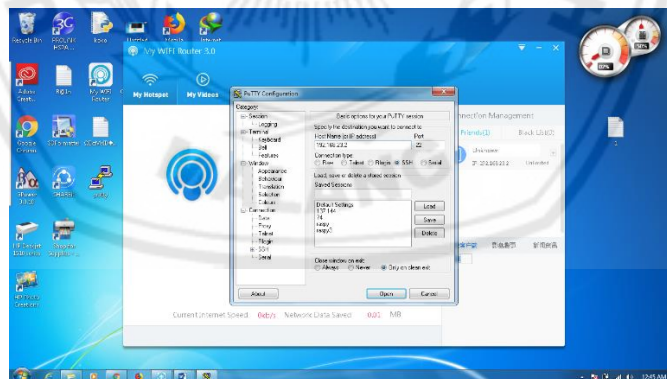
Pertama buka software My Wifi Router untuk mengaktifkan hotspot portable pada laptop



Gambar 3.6 My Wifi Router

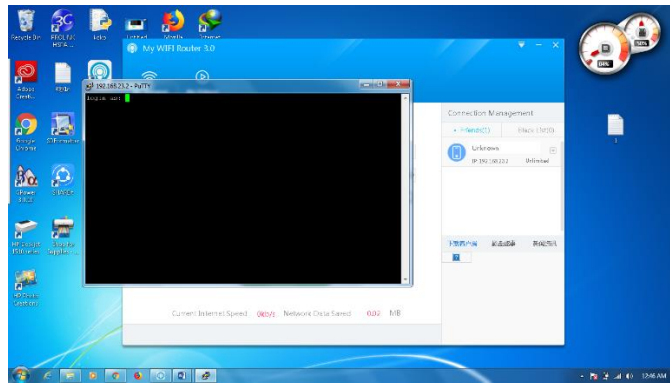
Setelah ngaktifkan hotspot portable maka nyalakan raspberry sehingga raspberry akan otomatis konek pada hotspot portable .tanda jika raspberry sudah konek akan terlihat ip address pada kolom sebelah kanan.

Langkah kedua buka software putty seperti gambar brikut:



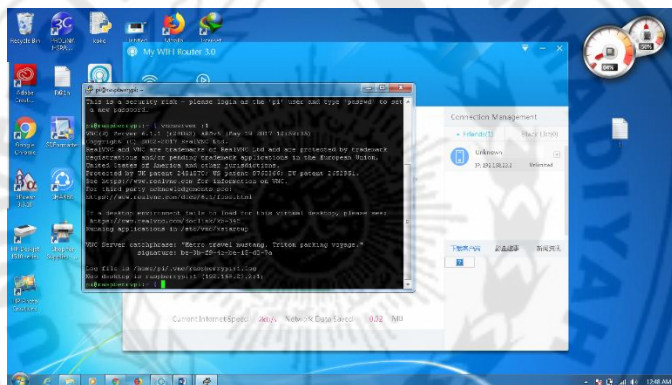
Gambar 3.7 Seting Putty

Setelah software terbuka masukkan ip address raspberry pi pada kolom host name dan host tidak usah di ganti, kemudian tekan load sehingga akan masuk pada Command Putty seperti gambar berikut:



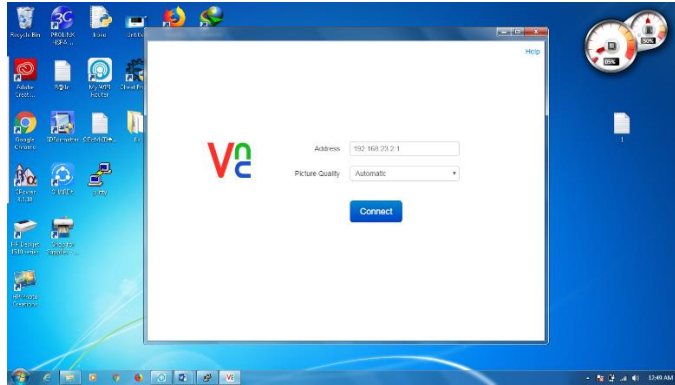
Gambar 3.8 Tampilan Cmd Raspbrry Pi

Disana disuruh memasukkan password raspberry untuk id masukkan “pi” sedangkan untuk password masukan “raspbrry”. kemudan untuk ketik “vncserver :1” untuk mengaktifkan vnc server pada raspbrry pi. Sperti gambar berikut:



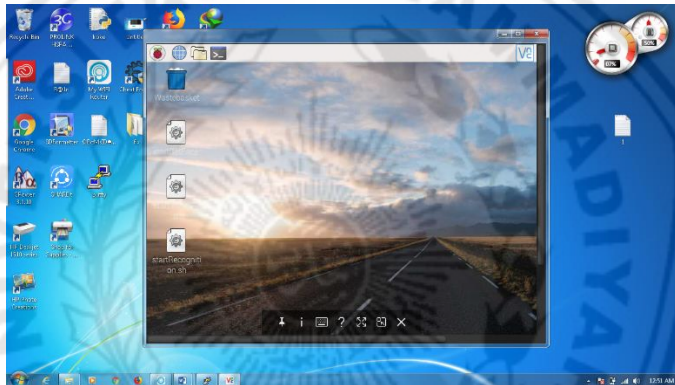
Gambar 3.9 Tampilan Cmd Raspbrry Pi

Langkah ke tiga setelah vnc server pada raspbrry diaktifkan kemudian buka vnc server pada pc. Kemudian masuk ip address vnc server “192.168.23.2:1” lalu tekan *connec.* sperti gambar berikut:



Gambar 3.10 Software VNC Server

Jika diminta password vnc server masukkan “raspberry” setelah memasukkan password tampilan akan masuk ke desktop raspberry.



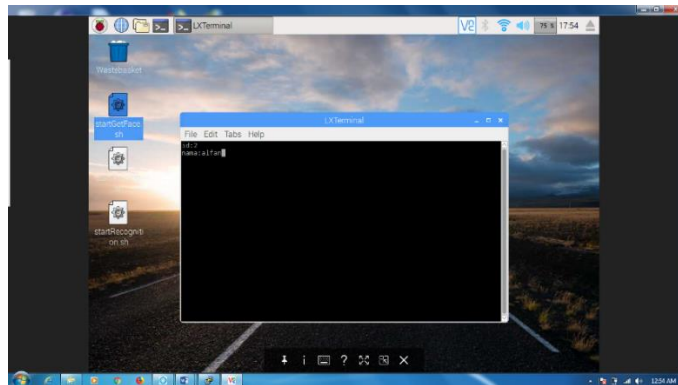
Gambar 3.11 Tampilan Desktop Raspberry Pi

Pada desktop raspberry pi terdapat tiga program utama yaitu: getface, training, dan recognition. getFace untuk memasukkan data mana, id, dan gambar wajah. Training untuk menggabungkan nama dengan database wajah yang kemudian akan disimpan pada file .yaml berupa data perhitungan. Recognition adalah proses pengenalan wajah.

3.3.2 Proses Pengambilan data Wajah

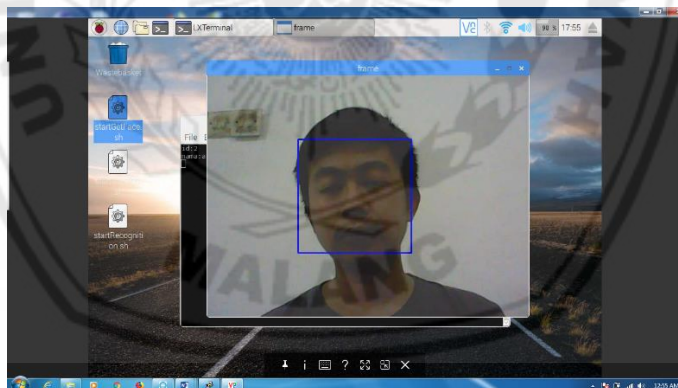
Berdasarkan rancangan *software* di atas maka akan dilakukan pengambilan data training yang akan digunakan sebagai data pembandingan yang nantinya akan disimpan di dalam *database*. Tahapan - tahapan pengambilan data training yaitu sebagai berikut :

Buka program getface dengan klik dua kali sehingga akan muncul seperti gambar berikut:



Gambar 3.12 Input Data Nama dan Id

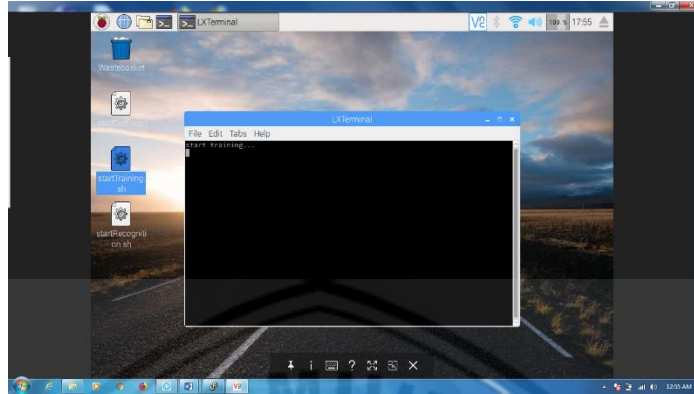
Disitu disuruh memasukkan “id” dan “nama”, untuk id dimasukkan angka untuk setiap orang id harus berbeda agar tidak terjadi salah deteksi. untuk nama kita masukkan terserah. Setelah memasukkan “nama” dan “id” kita akan langsung memasukkan gambar wajah orang yang akan kita simpan. Seperti gambar berikut:



Gambar 3.13 Input Data Wajah

Untuk pengambilan gambar akan tersimpan sebanyak 20 gambar. Setelah setelah menyimpan gambar sebanyak 20 sistem akan keluar sendiri.

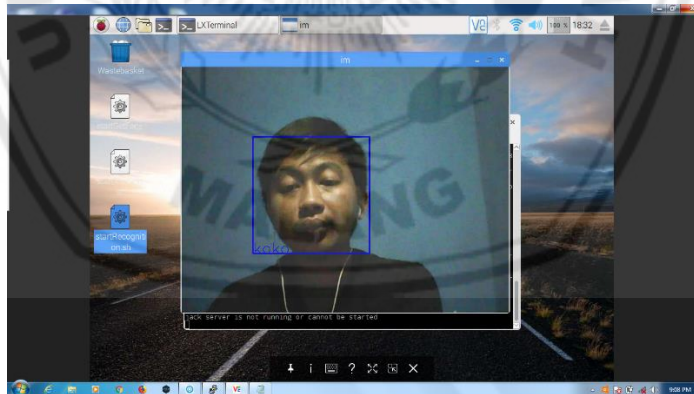
Selanjutnya buka program training disini program akan menggabungkan data wajah dengan mana. Tunggu proses training sampai selesai. Setelah selesai akan keluar sendiri. Seperti gambar berikut:



Gambar 3.14 Proses Training

3.3.3 Proses Pengenalan Wajah

Pertama buka program reconition .pada program ini akan mendeteksi wajah orang yang tersimpan di dalam data base.



Gambar 3.15 Proses *Face Recognition*

Jika wajah dikenali maka akan menampilkan tulisan nama pada bagian sisi bawah kotak dan akan mengeluarkan suara nama orang tersebut. Sementara jika wajah tidak dikenali maka akan menampilkan tulisan “unknow” dan mengeluarkan suara “id tidak ditemukan”.